AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (currently amended) A method for configuring a processing system that comprises program code, the program code comprising a plurality of instructions for processing data packets in a communications network, the method comprising:

~~receiving a program code, comprising a plurality of instructions for the communications network~~,

dividing the program code into a plurality of sequences, ~~that each perform~~ sequence performing a certain task on a data packet passing through the communications network,

providing a processor instruction memory comprising rows and columns;

allocating each sequence to a column of the processor instruction memory;

defining, based on the program code, a plurality of relocation objects, each corresponding to a dependency relationship between two or more of the sequences, and

by means of the relocation object providing information that there is an alternative sequence to jump to at the instruction at which the relocation object is located~~allocating the sequences to a processor instruction memory~~.

2. (previously presented) The method according to claim 1, comprising the steps of forming at least one directed graph, based on at least some of the sequences and at least some of the relocation objects, and determining a longest execution path through the directed graph.

3. (previously presented) The method according to claim 2, comprising the step of entering at least one state preserving operation in the instruction memory, so as to make at least two execution paths equally long.

4. (previously presented) The method according to claim 3, comprising the step of moving at least one sequence in the instruction memory.

5. (previously presented) The method according to claim 3, wherein the length of the at least two execution paths correspond to the longest execution path.

6. (previously presented) The method according to claim 1, comprising the step of determining the existence of any circle reference by any of the relocation objects between any of the sequences.

7. (previously presented) The method according to claim 1, comprising the step of linking at least one sequence, obtained by the step of dividing the program code, to a sequence, obtained by dividing another program code.

8. (currently amended) A processing system ~~including a computer-readable medium encoded with a computer program~~ for processing data packets in a communications network, the processing system comprising:

an assembler ~~adapted to receive~~comprising ~~a~~ program code, comprising a plurality of instructions for the communications network, wherein the assembler being adapted to:

divide the program code into a plurality of sequences, ~~that~~ each sequence perform a certain task on a data packet passing through the communications network,

define, based on the program code, a plurality of relocation objects ~~(10)~~, each corresponding to a dependency relationship between two or more of the sequences, ~~and~~

a processor instruction memory comprising rows and columns,

a linker being adapted to allocate each sequence ~~the sequences~~ to a column of the processor instruction memory,

wherein the relocation object provides information that there is an alternative sequence to jump to at the instruction at which the relocation object is located.

9. (previously presented) The processing system according to claim 8, wherein the assembler is adapted to form at least one directed graph, based on at least some of the sequences and at least some of the relocation objects, and the linker is adapted to determine a longest execution path through the directed graph.

10. (previously presented) The processing system according to claim 9, wherein the linker is adapted to enter at least one state preserving operation in the instruction memory, so as to make at least two execution paths equally long.

11. (previously presented) The processing system according to claim 10, wherein the linker is adapted to move at least one sequence in the instruction memory.

12. (previously presented) The processing system according to claim 10, wherein the length of the at least two execution paths correspond to the longest execution path.

13. (previously presented) The processing system according to claim 8, wherein the linker is adapted to determine the existence of any circle reference by any of the relocation objects between any of the sequences.

14. (previously presented) The processing system according to claim 8, wherein the linker is adapted to link at least one sequence, obtained by dividing the program code, to a sequence, obtained by dividing another program code.